

OMS FDIR - Initial Prototyping

Eric W. Taylor and Matthew A. Hanson
Ford Aerospace Corporation

ABSTRACT

The Space Station Freedom Program (SSFP) Operations Management System (OMS) will automate major management functions which coordinate the operations of onboard systems, elements and payloads. The objectives of OMS are to improve safety, reliability and productivity while reducing maintenance and operations cost. This will be accomplished by using advanced automation techniques to automate much of the activity currently performed by the flight crew and ground personnel. OMS requirements have been organized into five task groups: 1) Planning, Execution and Replanning, 2) Data Gathering, Preprocessing and Storage, 3) Testing and Training, 4) Resource Management, and 5) Caution & Warning and Fault Management for onboard subsystems. The scope of this prototyping effort falls within the Fault Management requirements group. The prototyping will be performed in two phases. Phase 1 is the development of an onboard communications network FDIR system. Phase 2 will incorporate global FDIR for onboard systems. Research into the applicability of expert systems, object-oriented programming, fuzzy sets, neural networks and other advanced techniques will be conducted. This paper discusses the goals and technical approach for this new SSFP research project.

INTRODUCTION

The Space Station Freedom Program (SSFP) Operations Management System (OMS) is a set of functions which includes application software and manual interactions with the Freedom Station either from onboard or on the ground. OMS requirements have been organized into five task groups: 1) Planning, Execution and Replanning, 2) Data Gathering, Preprocessing and Storage, 3) Testing and Training, 4) Resource Management, and 5) Caution & Warning and Fault Management for onboard subsystems. The scope of this prototyping effort falls within the Caution & Warning (C&W) and Fault Management requirements group. The purpose is to address the global fault detection, isolation, and reconfiguration (FDIR) requirements for OMS automation within the Space Station Freedom program.

Initial prototyping will concentrate on Network Management FDIR, a user interface, and a modular system architecture which incorporates advanced automation (i.e. expert systems, neural networks, etc). Subsequent prototyping will integrate the initial prototype with other prototype activities which are currently being developed in the SSFP End-to-end Test Capability (ETC) testbeds.

SSFP OMS BACKGROUND

The Command and Control Architecture for Integrated Operations Control, as defined in [PDDR88], specifies a hierarchical command structure with multiple tiers. Two levels are defined and are called Tier I and Tier II. "Tier I shall be the source of real-time and near real-time command and control to Tier II and shall provide top-level management and station-wide integration." "Tier II shall receive and execute commands from Tier I and provide the required information to

it." [PDDR88]

The OMS application software consists of an onboard portion which is designated as the Operations Management Application (OMA); and a ground resident portion which is designated as the Operations Management Ground Application (OMGA) [PDDR88]. The OMA is the onboard portion of the Tier I Command and Control Architecture. As such, the OMA interfaces with and provides top level command and control to the Tier II Distributed System Executives, Element Managers, and Payload Managers for integrated real-time and near real-time operations. Tier II, in turn, supports the OMA by providing management of their sub-components and by providing health, status, and other information, as required, to the OMA.

The OMGA is the ground application software portion of the Tier I Integrated Operations Control of the OMS. The OMGA is the only command access from the ground to the OMA. It performs unique functions in support of ground activities for operations, complements OMA functions in support of operations, and monitors OMA Activities [PDDR88].

In general, the Tier II managers are responsible for their own FDIR. The OMA performs station-wide FDIR when more than one Tier II manager is affected. The OMGA augments the OMA's actions for station-wide FDIR. Once the C&W function annunciates a failure or anomaly, the FDIR function identifies the source of the failure and the necessary actions to be taken. [PDDR88]

TECHNICAL APPROACH

Initially, prototype functionality will be targeted for onboard communication network FDIR. This portion of the prototype represents a Tier II Executive. It will communicate with network management entities within the internet to provide a central point of network management control. The Tier I OMA FDIR will communicate with this Tier II Network Executive providing the integrated onboard network FDIR aspect of Station-Wide Fault Management (SWFM).

The ISO model for Open Systems Communications defines services and protocols which provide for the exchange of network management information between Network Managers and Agents. MAP/TOP is a specification of the ISO standards which define this communication between Network Managers. Currently, these functions are not completely specified or available but it is assumed that the ISO standards (and possibly the MAP/TOP specification) will be used on the SSFP communications networks. We intend to use these types of functions to exchange information between the Tier II Network Executive and other network managers within the internet to perform communications network fault management.

When this is on solid ground, the second phase of this prototyping effort will develop a Tier I Manager which initially communicates with the above Tier II Network Executive. Subsequently, the prototype will include Station-Wide Fault Management, communicating with subsystem prototypes being developed on related SSFP prototyping projects (e.g. EPS, ECLSS, GN&C, OMS Short Term Planning, etc.).

Two prototype systems are currently envisioned: an OMA system residing in the JSC DMS test bed, and an OMGA system residing in the SSCC test bed. Towards the end of this project, end-to-end demonstration scenarios are planned within this portion of the End-to-end Test Capability (ETC) environment.

ADVANCED AUTOMATION IN OMS FDIR

The goal of this prototype effort is to exhibit the utility of knowledge based systems and other advanced automation techniques in Space Station Freedom program FDIR. For example, production systems may be employed during fault isolation and recovery where expert knowledge of system-wide faults can be utilized. Object Oriented Programming (OOP) and Frames may be used to create and maintain a model of the network (both at Tier I and Tier II levels) which will aid in the fault isolation phase. Neural Networks and Fuzzy Logic applications will be explored as the project progresses.

A knowledge based system will contain the knowledge required to perform fault isolation and recovery given fault and status messages from systems under its influence. These messages include Caution and Warning and will be collected by the FDIR system either solicited or unsolicited. The system will proceed to determine the "root" fault from among "side-effect" faults from these messages, reaching a conclusion from the available information or initiating tests to acquire additional information.

Expert knowledge will be used to determine which faults to pursue and which tests to initiate. Which of the faults (if multiple), is most severe, thus requiring immediate attention? Which test will most likely return the most informative data at the least expense?

Faults which appear, and subsequently clear will be tracked and chronologically logged, building a database of problem areas. This information can then be used during later FDIR functions. All information required to make a correct diagnosis will not always be available, so the system will be able to deduce from incomplete data.

The human OMS Operator will not be subjected to large volumes of "raw" alarm data. Recovery procedures dealing with faults that have been determined as not requiring operator intervention will be performed automatically. Many recovery procedures will require the assistance of flight crew or ground support personnel, such as the off-line test of a piece of equipment or the swap or repair of a board or component. Other recovery procedures could recommend contacting the responsible vendor. After a recovery procedure has been followed, the FDIR system will initiate tests on the affected system/component before the fault instance is "closed".

SOFTWARE ENVIRONMENT

There are three major aspects to the prototype's software development: 1) the knowledge-based modules, 2) the user interface, and 3) the underlying system code. "Underlying system code" refers to the code that integrates the user interface, knowledge-based modules, operating system, network functions, etc. In short, it includes all code except the user interface and the knowledge-based modules.

Some basic assumptions have been made about the software environment of this effort as follows:

- o Unix (or Posix) operating system
- o X-windows application/user interface.
- o ISO network standards

While some non-Unix options are considered for early prototyping proof-of-concept, Unix is the ultimate target operating system environment.

KNOWLEDGE-BASED SYSTEM

Two production systems were considered for this prototyping effort: CLIPS and Ops83. CLIPS was developed by JSC MPAD, and Ops83 was developed by Charles Forgy's Production Systems Technologies. Ops83 has proven to be the fastest production system benchmarked, and it contains a full-featured procedural language for top-level production system control. In addition, Ops83 offers full control of the recognize-act cycle and readily interfaces to C, Ada and Fortran language modules. Procurement of Ops83 has been initiated for this project.

USER INTERFACE

The Transportable Application Environment (TAE) is a user interface development

tool which runs under Unix and X-Windows. It provides the user interface developer with an interactive tool to create windowing hierarchies with associated menus, buttons, icons, slide bars, animated graphics, etc. The TAE system generates "C" code which references X-Window's Xlib. The developer then incorporates his application code into the code generated by TAE. This system will be used to efficiently generate the FDIR system user interface(s).

UNDERLYING CODE

The "C" programming language will be used for initial prototyping. The C++ Object Oriented Programming Language will be used and further applicability will be explored. As the project progresses, the Ada language will be used as it is the required language of the Space Station Freedom Program.

HARDWARE ENVIRONMENT

A secondary goal of this project is to demonstrate OMGA and OMA FDIR functionality in the ETC environment. This implies hosting in both the SSCC test bed (for OMGA) and the DMS test bed (for OMA). Initial prototyping will be carried out in the SSCC test bed for both systems. Selected software methodologies should make any ports to the DMS test bed insignificant efforts. Sun (Unix) workstations are available in the SSCC test bed for this project, and all supportive software (TAE Plus, C++, and Ops83) has been (or is being) acquired.

SUMMARY

An overview of our initial approach to this prototyping effort was given. The first area for prototyping is advanced automation in communications network management. This will be approached assuming the OSI Reference Model will be employed on Space Station Freedom communication networks. The second area of prototyping will be to integrate the network management prototype into the OMS testbed environment. Some background on the OMS environment was given. This second effort will require extensive research into the current testbed configuration and capabilities. In addition, open lines of communication between ourselves and the testbed players will need to be more fully established.

REFERENCES

- [X3T987] ASC Technical Committee, "FDDI Station Management (SMT)", Draft American National Standard X3T9/85-X3T9.5/84-49 REV-2.1 APR 87
- [BAYE88] Bayer, Steven E., Harris, Richard A., et al, "A Review of Space Station Freedom Program Capabilities for the Development and Application of Advanced Automation" MTR-88D00059, MITRE Corp. for NASA/JSC, Houston, TX, DEC 88.
- [BERG89] Berger, Eugene L., "Concept of Operations for Network Management of Space Station Freedom Onboard Networks" WP-88D0041, MITRE Corp. for NASA/JSC, Houston, TX, MAR 89.
- [HUNT89] Huntington, Jill A., "OSI-based net management: Is It Too Early, or Too Late?", DATA COMMUNICATIONS, McGraw-Hill, Vol. 18, Number 3, MAR 89, pp. 111-129.
- [PIET88] Pietras, J., McCoy, E., et, "Concept of Operations for Space Station Information System (SSIS) Network Integration Management, Part 1: Architecture and Functions", WP-88W00268, MITRE Corp. for NASA/JSC, Houston, TX, OCT 88.
- [NOLE88] Noles, James, McCoy, Emily, et al, "A Network Services Reference

Configuration for the Space Station Information System", WP-88W00139, MITRE Corp for NASA/JSC, Houston, TX, OCT 88.

- [CRON88] Cronk, Robert N., Callahan, Paul H., et al, "Rule-Based Expert Systems for Network Management and Operations: An Introduction" IEEE Network, IEEE Publishing Services, SEP 88, pp. 7-21.
- [SUTT88] Sutter, Mark T., Zeldin, Paul E., "Designing Expert Systems for Real-Time Diagnosis of Self-Correcting Networks" IEEE Network, IEEE Publishing Services, SEP 88, pp. 43-51.
- [MAPT87] MAP/TOP User's Group, "Technical and Office Protocol Specifications" Society of Manufacturing Engineers, Version 3.0, Draft, Implementation Release, MAY 87.
- [PDDR88] Space Station Projects Office, "Space Station Projects Description and Requirements Document, Vol. 3: Projects Design and Development Requirements," JSC31000 Rev. D, pp. 159-194.
- [ECKL89] Eckelkamp, Rick, et. al., "OMS Evolution Plan," Johnson Space Center, JAN 1989
- [HCIG88] "Space Station Freedom Program Human-Computer Interface Guide," Version 2.1, DEC 1988

